Applications of Legendre-Fenchel transformation to computer vision problems

Ankur Handa, Richard A. Newcombe, Adrien Angeli, Andrew J. Davison

Department of Computing,
Huxley Building,
180, Queens Gate,
SW7 2AZ, London
{ahanda,r.newcombe08,a.angeli,ajd}@imperial.ac.uk,
WWW home page: http://www.doc.ic.ac.uk/∼ahanda

Abstract. We aim to provide a small background on Lengenre-Fenchel transformation, the applications of which have been increasingly getting popular in computer vision. A general motivation follows up with standard examples. Then we take a good view on their applications in solving various standard computer vision problems e.g. image denoising, optical flow, image deconvolution etc.

Keywords: Legendre-Fenchel transformation, Convex functions, Optimisation

1 Legendre-Fenchel transformation

Legendre-Fenchel (LF) transformation of a **continuous** but not necessarily differentiable function $f : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$, is defined as

$$f^*(p) = \sup_{x \in \mathbb{R}} \{px - f(x)\} \tag{1}$$

Geometrically it means that we are interested in finding a point x on the function $f(x)$ such that the slope of line p passing through $(x, f(x))$ has a maximum intercept on the y-axis. This happens to be the point on the curve which has a slope p which is nothing but the tangent at that point.

$$p = f'(x) \tag{2}$$

A vector definition can be written as

$$f^*(\mathbf{p}) = \sup_{\mathbf{x} \in \mathbb{R}^n} \{\mathbf{x}^t \mathbf{p} - f(\mathbf{x})\} \tag{3}$$

2 Motivation: Why do we need it?

2.1 Duality

Duality is the principle of looking at a function or a problem from two different perspectives namely the primal and the dual form. When working in optimisation theory, often and in general a deep understanding of a given function

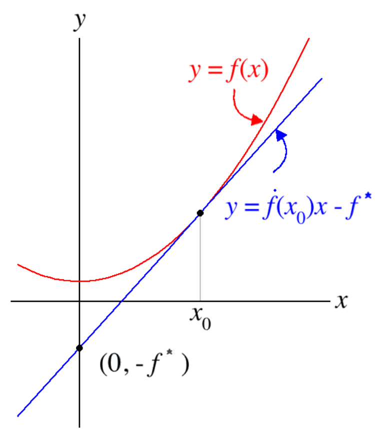

Fig. 1. Image Courtesy: Wikipedia.org

is required. For instance, one would like to know whether a given function is linear, whether the function is well behaved in a given domain etc. to name a few. Transformations are one way of mapping the function to another space where better and easy ways of understanding the function emerge out. Take for instance Fourier Transform and Laplace Transform. Legendre-Fenchel is one such transform which maps the $(x, f(x))$ space to the space of slope and conjugate that is $(p, f^*(p))$. However, while the Fourier transform consists of an integration with a kernel, the Legendre Fenchel transform uses *supremum* as the transformation procedure. Under the assumption that the transformation is reversible, one form is the *dual* of the other. This is easily expressed as

$$(x, f(x)) \Longleftrightarrow (p, f^*(p)) \tag{4}$$

p is the slope and $f^*(p)$ is called the convex conjugate of the function $f(x)$. A conjugate allows one to build a dual problem which may be easier to solve than the primal problem. Legendre-Fenchel conjugate is always convex.

3 How to use the Duality

There are two ways of viewing a curve or a surface, either as a locus of points or envelope of tangents [1]. Now, let us imagine that we want to use the duality of tangents to represent a function $f(x)$ instead of points. A tangent is parameterised by two variables namely the slope p and the intercept it cuts on negative y-axis (using negative y-axis for intercept is purely a matter of choice) which is denoted by c. We provide two ways to solve for the intercept and the slope and arrive at the same result.

3.1 Motivation I

Let us denote the equation of line having a slope p and intercept c by

$$y = px - c \tag{5}$$

Now imagine this line is to touch the function $f(x)$ at x, then we can equate both of them and write as

$$px - c = f(x) \tag{6}$$

Also suppose that the function $f(x)$ is convex and is x^2, a parabola (as an example). Then we can solve for quadratic equation

$$x^2 - px + c = 0 \tag{7}$$

Now this quadratic equation has two roots

$$x = \frac{p \pm \sqrt{p^2 - 4c}}{2} \tag{8}$$

But we know that this line should touch this convex function only at once (if the function was non-convex, the line could touch the function at two points) and because we want to use this line as a tangent to represent the function in dual space. Therefore, the roots of this equation should be equal which is to say that the determinant of the above quadratic equation should be zero i.e. $p^2 - 4c = 0$ which gives us

$$f^*(p) = c = \frac{p^2}{4} \tag{9}$$

This is nothing but our Legendre-Fenchel transformed convex conjugate and

$$p = 2x \tag{10}$$

is the slope. That is

$$(x, f(x)) \Longleftrightarrow (p, f^*(p)) \tag{11}$$

$$(x, x^2) \Longleftrightarrow (p, \frac{p^2}{4}) \tag{12}$$

3.2 Motivation II

For we know that

$$y = px - c \tag{13}$$

to be a tangent to $f(x)$ at x it must be that

$$p = f'(x) \tag{14}$$

Again, if the function is

$$f(x) = x^2 \tag{15}$$

we can take the first order derivative to obtain the slope which is

$$p = f'(x) = 2x \tag{16}$$

$$x = f'^{-1}(p) = \frac{p}{2} \tag{17}$$

Replacing y we get

$$y = f(x) \tag{18}$$

and substituting x we get,

$$y = f(f'^{-1}(p)) \tag{19}$$

$$y = f(\frac{p}{2}) = \frac{p^2}{4} \tag{20}$$

and therefore we can solve for c as

$$c = pf'^{-1}(p) - f(f'^{-1}(p)) \tag{21}$$

$$c = p\frac{p}{2} - \frac{p^2}{4} \tag{22}$$

$$c = \frac{p^2}{4} \tag{23}$$

4 How about fuctions which are not differentiable everywhere?

Let us now turn our attention towards the cases where a function may not be differentiable at a given point x^* and has a value $f(x^*)$. In this case we can rewrite our equation as

$$px^* - c = f(x^*) \tag{24}$$

which means

$$c = px^* - f(x^*) \tag{25}$$

is only a linear function of p. So the duality of a non-differentiable point at x induces a linear function of slope in the dual space. And p can take a value from $[f'(x^*_-),f'(x^*_+)]$. This is because the slope of the tangent at a point in the very small vicinity of x^* to the left side of it, denoted by x^*_- is $f'(x^*_-)$ and the right,

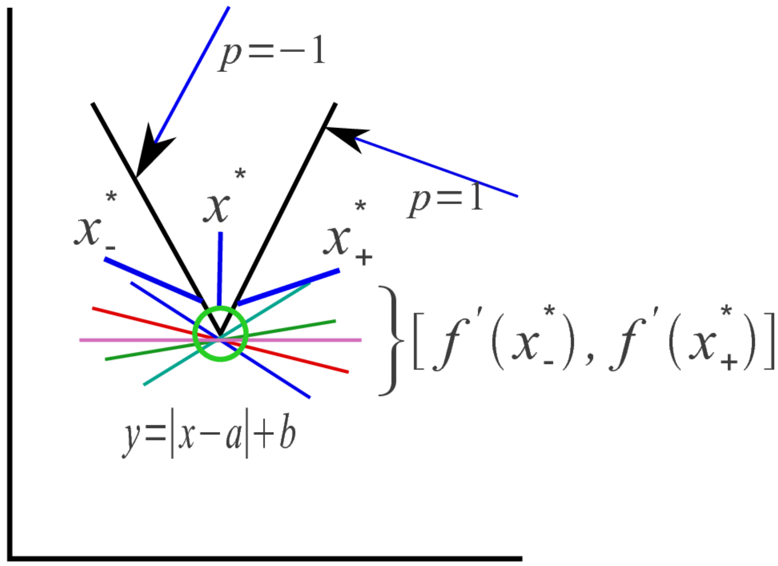

Fig. 2. At the point x^* we can draw as many tangents we want with slopes ranging from $[-1, 1]$ and they form the subgradients of the curve at x^*.

denoted by x^*_+ is $f'(x^*_+)$. At the point of discontinuity in the space of function $f(x)$, we can draw many tangents with slopes ranging from $[f'(x^*_-), f'(x^*_+)]$. This interval is defined as the *subgradient*. This is explained briefly in the next section.

Therefore, the point of non-differentiability in the *primal* space can be readily explained in dual space with a continuously varying slope in the range $[f'(x^*_-), f'(x^*_+)]$ defining a linear function, $c(p)$, of this slope. Therefore, even if the *primal* space is non-differentiable, the *dual* space is not.

4.1 Subdifferential and Subgradient

In calculus, we are, majority of time, interested in minimising or maximising a function f. The point $\hat{x}$ which minimises the function is referred to as the critical point, the minimiser of function i.e. $\nabla f(\hat{x}) = 0$. Convex functions belong to the class of functions which have global minimiser. However there are certain convex functions which are not differentiable everywhere therefore one can not compute the gradient. A notorious function $f(x) = |x|$ is an example of convex function which is not differentiable at $x = 0$. Instead, one defines the *subdifferential*. A *subdifferential* is therefore formalised as $\partial f(x)$ such that

$$\partial f(x) = \{y \in \mathbb{R}^n : \langle y, x' - x\rangle \leq f(x') - f(x), \forall x' \in \mathbb{R}^n\} \tag{26}$$

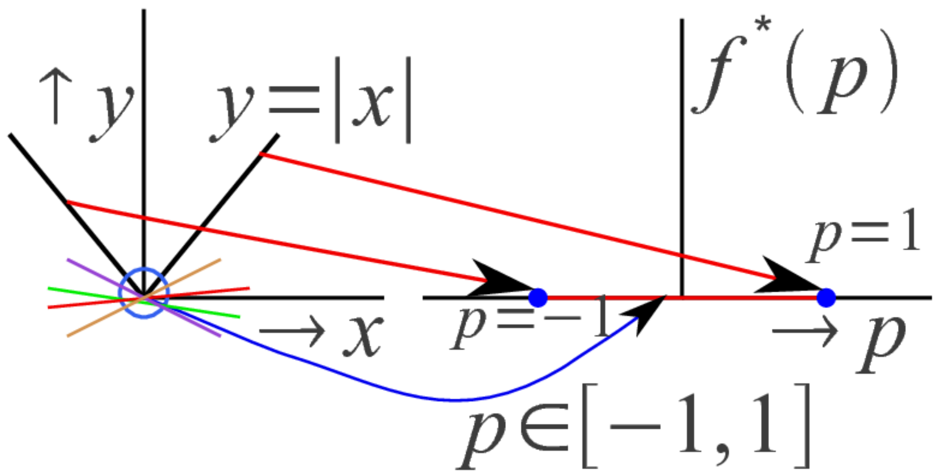

Fig. 3. An illustration of duality where lines are mapped to points while points are mapped to lines in the dual space.

In simpler term, a *subdifferential* is defined as the slope of the line at point x such that it is either always touching or remaining below the graph of function. For the same notorious $|x|$ function the differential is not defined at $x = 0$ because $\frac{x}{|x|}$ is not defined at $x = 0$ while *subdifferential* at point $x = 0$ is the close interval [-1, 1] because we can always draw a line with a slope between [-1,1] which is always below the function. The subdifferential at any point $x < 0$ is the singleton set {-1}, while the subdifferential at any point $x > 0$ is the singleton {1}. Members of the *subdifferential* are called *subgradients*.

4.2 Proof: Legendre-Fenchel conjugate is always convex

$$f^*(\mathbf{z}) = \sup_{\mathbf{x}\in\mathbb{R}^n} \{\mathbf{x}^t\mathbf{z} - f(\mathbf{x})\} \tag{27}$$

In order to prove that function is convex we need to prove that for a given $0 \leq \lambda \leq 1$ the function should obey Jensen's inequality i.e.

$$f^*(\lambda\mathbf{z}_1 + (1-\lambda)\mathbf{z}_2) \leq \lambda f^*(\mathbf{z}_1) + (1-\lambda)f^*(\mathbf{z}_2) \tag{28}$$

Let us expand the left hand side of the inequality to be proved.

$$f^*(\lambda\mathbf{z}_1 + (1-\lambda)\mathbf{z}_2) = \sup_{\mathbf{x}\in\mathbb{R}^n} \{\mathbf{x}^t(\lambda\mathbf{z}_1 + (1-\lambda)\mathbf{z}_2) - f(\mathbf{x})\} \tag{29}$$

We can rewrite $f(x)$ as

$$f(\mathbf{x}) = \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{x}) \tag{30}$$

and replacing it in the equation above yields a new expression which is

$$f^*(\lambda\mathbf{z}_1 + (1-\lambda)\mathbf{z}_2) = \sup_{\mathbf{x}\in\mathbb{R}^n}\{\lambda(\mathbf{x}^t\mathbf{z}_1 - f(\mathbf{x})) + (1-\lambda)(\mathbf{x}^t\mathbf{z}_2 - f(\mathbf{x}))\} \tag{31}$$

But we know that

$$\sup_{\mathbf{x}\in\mathbb{R}^n}\{\lambda(\mathbf{x}^t\mathbf{z}_1 - f(\mathbf{x})) + (1-\lambda)(\mathbf{x}^t\mathbf{z}_2 - f(\mathbf{x}))\} \leq \sup_{\mathbf{x}\in\mathbb{R}^n}\{\lambda(\mathbf{x}^t\mathbf{z}_1 - f(\mathbf{x}))\} + \sup_{\mathbf{x}\in\mathbb{R}^n}\{(1-\lambda)(\mathbf{x}^t\mathbf{z}_2 - f(\mathbf{x}))\} \tag{32}$$

It is the property of *supremum* which states that

$$\sup\{\mathbf{x}+\mathbf{y}\} \leq \sup\{\mathbf{x}\} + \sup\{\mathbf{y}\} \tag{33}$$

Therefore, we can substitute for

$$\sup_{\mathbf{x}\in\mathbb{R}^n}\{\lambda(\mathbf{x}^t\mathbf{z}_1 - f(\mathbf{x}))\} = \lambda f^*(\mathbf{z}_1) \tag{34}$$

and

$$\sup_{\mathbf{x}\in\mathbb{R}^n}\{(1-\lambda)(\mathbf{x}^t\mathbf{z}_2 - f(\mathbf{x}))\} = (1-\lambda) f^*(\mathbf{z}_2) \tag{35}$$

We arrive it our desired result which is

$$f^*(\lambda\mathbf{z}_1 + (1-\lambda)\mathbf{z}_2) \leq \lambda f^*(\mathbf{z}_1) + (1-\lambda) f^*(\mathbf{z}_2) \tag{36}$$

Thefore, $f^*(z)$ is convex always irrespective of the whether the function $f(x)$ is convex or not.

5 Examples

5.1 Example 1: Norm function (non-differentiable at zero)

$$f(y) = ||y|| \tag{37}$$

$$f^*(z) = \sup_{y\in\mathbb{R}}\{y^t z - ||y||\} \tag{38}$$

By using Cauchy-Schwarz inequality we can also write

$$||y|| = \max_{||b||\leq 1} y^t b \tag{39}$$

Now, we know that the maximum value of $y^t b$ is $||y||$ so it is trivial to see that

$$\max_{||b||\leq 1}\{y^t b - ||y||\} = 0 \quad \forall y \in \mathbb{R} \tag{40}$$

Therefore, we can write the conjugate as

$$f^*(z) = \begin{cases} 0 & \text{if } ||z|| \leq 1 \\ \infty & \text{otherwise} \end{cases}$$

The fact that the conjugate is ∞ when $||z|| > 1$ can be easily explained using Figure 4.

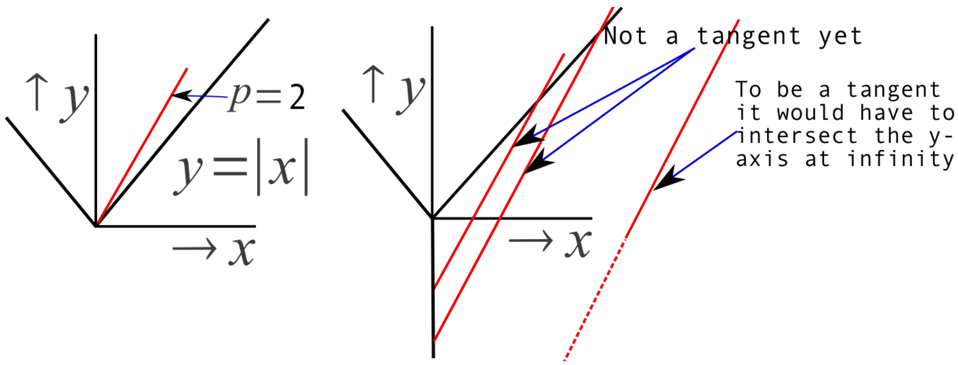

Fig. 4. The image explains the process involved when fitting a tangent with slope $p = 2$ or $z = 2$. Any line with slope $\notin$ [-1,1] has to intersect the y-axis at ∞, to be able to be tangent to the function.

5.2 Example 2: Parabola

$$f(y) = y^2 \tag{41}$$

LF transform of a function $f(y)$ for an n-dimensional vector y is defined as

$$f^*(z) = \sup_{y \in \mathbb{R}} \{yz - f(y)\} \tag{42}$$

The function attains maxima when

$$\partial_y (yz - y^2) = 0 \tag{43}$$

$$z - 2y = 0 \tag{44}$$

$$z = 2y \tag{45}$$

substituting the value of y in the above function $f^*(z)$ we get

$$f^*(z) = z\frac{z}{2} - \left(\frac{z}{2}\right)^2 = \frac{1}{4}z^2 \tag{46}$$

5.3 Example 3: A general quadratic n-D curve

$$f(\mathbf{y}) = \frac{1}{2}\mathbf{y}^t A \mathbf{y} \tag{47}$$

Let us assume that A is symmetric, then the LF transform of a function $f(y)$ for an n-dimensional vector y is defined as

$$f^*(\mathbf{z}) = \sup_{\mathbf{y} \in \mathbb{R}^n} \{\mathbf{y}^t \mathbf{z} - f(\mathbf{y})\} \tag{48}$$

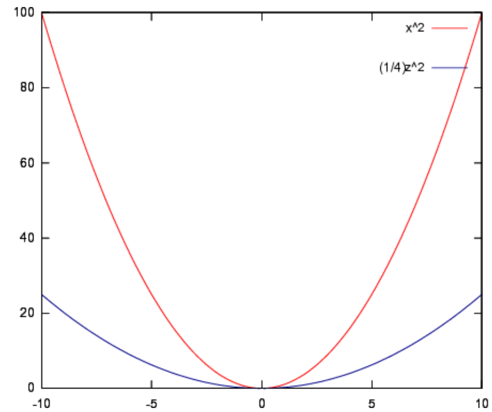

Fig. 5. The plot shows the parabola y^2 and it's conjugate which is also a parabola $\frac{1}{4}z^2$.

The function attains maxima when

$$\partial \mathbf{y}(\mathbf{y}^t\mathbf{z} - \frac{1}{2}\mathbf{y}^t A\mathbf{y}) = 0 \tag{49}$$

$$\mathbf{z} - \frac{1}{2}(A + A^T)\mathbf{y} = 0 \tag{50}$$

$$\mathbf{z} - A\mathbf{y} = 0 \quad [\because A \quad \text{is symmetric}] \tag{51}$$

$$\mathbf{y} = A^{-1}z \tag{52}$$

substituting the value of y in the above function $f^*(z)$ we get

$$f^*(\mathbf{z}) = (A^{-1}\mathbf{z})^t\mathbf{z} - \frac{1}{2}(A^{-1}\mathbf{z})^t A(A^{-1}\mathbf{z}) \tag{53}$$

$$= \mathbf{z}^t A^{-1}\mathbf{z} - \frac{1}{2}\mathbf{z}^t A^{-1}AA^{-1}\mathbf{z} \tag{54}$$

$$= \mathbf{z}^t A^{-1}\mathbf{z} - \frac{1}{2}\mathbf{z}^t A^{-1}AA^{-1}\mathbf{z} \tag{55}$$

$$= \frac{1}{2}\mathbf{z}^t A^{-1}\mathbf{z} \tag{56}$$

5.4 Example 4: l^p Norms

$$f(y) = \frac{1}{p}||y||^p \quad \forall \quad 1 < p < \infty \tag{57}$$

Again writing the LF transform as

$$f^*(z) = \sup_{y \in \mathbb{R}}\{y^t z - \frac{1}{p}||y||^p\} \tag{58}$$

$$\partial_y(y^t z - \frac{1}{p}||y||^p) = 0 \tag{59}$$
$$z - ||y||^{p-1}\frac{y}{||y||} = 0 \tag{60}$$
$$z = ||y||^{p-2}y \tag{61}$$
$$||z|| = ||y||^{p-1} \tag{62}$$
$$||y|| = ||z||^{\frac{1}{p-1}} \tag{63}$$
$$y = \frac{z}{||z||^{\frac{p-2}{p-1}}} \tag{64}$$

substituting this value of y into the function gives

$$f^*(z) = \frac{z^t}{||z||^{\frac{p-2}{p-1}}}z - \frac{1}{p}||z||^{\frac{p}{p-1}} \tag{65}$$
$$= \frac{z^t z}{||z||^{\frac{p-2}{p-1}}} - \frac{1}{p}||z||^{\frac{p}{p-1}} \tag{66}$$
$$= \frac{||z||^2}{||z||^{\frac{p-2}{p-1}}} - \frac{1}{p}||z||^{\frac{p}{p-1}} \tag{67}$$
$$= ||z||^{2-\frac{p-2}{p-1}} - \frac{1}{p}||z||^{\frac{p}{p-1}} \tag{68}$$
$$= ||z||^{\frac{2(p-1)-(p-2)}{p-1}} - \frac{1}{p}||z||^{\frac{p}{p-1}} \tag{69}$$
$$= ||z||^{\frac{2p-2-p+2}{p-1}} - \frac{1}{p}||z||^{\frac{p}{p-1}} \tag{70}$$
$$= ||z||^{\frac{p}{p-1}} - \frac{1}{p}||z||^{\frac{p}{p-1}} \tag{71}$$
$$= (1 - \frac{1}{p})||z||^{\frac{p}{p-1}} \tag{72}$$
$$= (1 - \frac{1}{p})||z||^{\frac{1}{1-\frac{1}{p}}} \tag{73}$$
$$= \frac{1}{\frac{1}{(1-\frac{1}{p})}}||z||^{\frac{1}{1-\frac{1}{p}}} \tag{74}$$
$$\tag{75}$$

Let us call

$$q = \frac{1}{(1 - \frac{1}{p})} \tag{76}$$

Therefore, we obtain

$$f^*(z) = \frac{1}{q}||z||^q \tag{77}$$

$$\frac{1}{p} + \frac{1}{q} = 1 \tag{78}$$

5.5 Example 5: Exponential function

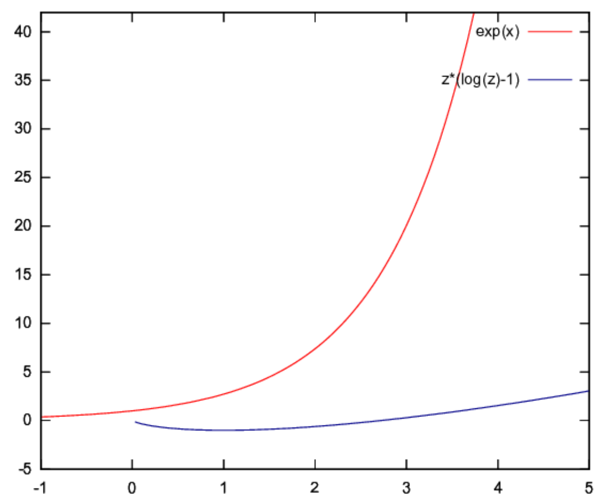

Fig. 6. The plot shows the function e^x and it's conjugate which is $z(\ln z - 1)$.

$$f(y) = e^y \tag{79}$$

$$f^*(z) = \sup_{y \in \mathbb{R}} \{y^t z - e^y\} \tag{80}$$

$$\tag{81}$$

if z < 0 : $\sup_{y \in \mathbb{R}} \{y^t z - e^y\}$ is unbounded so $f^*(z) = \infty$
if z > 0 : $\sup_{y \in \mathbb{R}} \{y^t z - e^y\}$ is bounded and can be computed as

$$\partial_y (yz - e^y) = 0 \tag{82}$$

$$z - e^y = 0 \tag{83}$$

$$y = \ln z \tag{84}$$

substituting the value of y in the above function $f^*(z)$ we get

$$f^*(z) = z \ln z - z = z(\ln z - 1) \tag{85}$$

if z = 0 : $\sup_{y\in\mathbb{R}}\{y^t z - e^y\} = \sup_{y\in\mathbb{R}}\{-e^y\} = 0$

5.6 Example 6: Negative logarithm

$$f(y) = -\log y \tag{86}$$

$$f^*(z) = \sup_{y\in\mathbb{R}}\{y^t z - (-\log y)\} \tag{87}$$

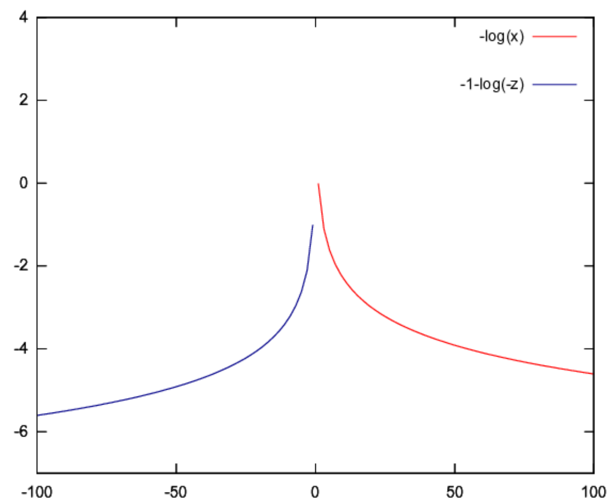

Fig. 7. The plot shows the function -log y and it's conjugate which is -1 -log (-z).

$$\partial_y(yz + \log y) = 0 \tag{88}$$

$$z + \frac{1}{y} = 0 \tag{89}$$

$$y = -\frac{1}{z} \tag{90}$$

Substituting this value back into the equation we get

$$f^*(z) = \frac{1}{-z}z + \log(1/-z) \tag{91}$$

$$f^*(z) = -1 - \log(-z) \tag{92}$$

This is only valid if z $<$ 0

6 Summary of noteworthy points

- The Legendre-Fenchel transform only yields convex functions
- Points of function f are transformed into slopes of f^*, and slopes of f are transformed into points of f^*
- The Legendre-Fenchel transform is more general than the Legendre transform because it is also applicable to non-convex functions as well as non-differentiable functions. The Legendre-Fenchel transform reduces to Legendre transform for convex functions.

7 Applications to computer vision

Many problems in computer vision can be expressed in the form of energy minimisations [7]. A general class of the functions representing these problems can be written as

$$\min_{x \in X} \{F(Kx) + G(x)\} \tag{93}$$

where F and G are proper convex functions and $K \in \mathbb{R}^{n \times m}$. Usually, $F(Kx)$ corresponds to regularisation term of the form $||Kx||$ and $G(x)$ corresponds to the data term. The dual form can be easily derived by replacing $F(Kx)$ with it's convex conjugate, that is

$$\min_{x \in X} \max_{y \in Y} \{\langle Kx, y \rangle - F^*(y) + G(x)\} \tag{94}$$

because F is a convex function then, by definition of Legendre-Fenchel transformation

$$F(Kx) = \max_{y \in Y} \{\langle Kx, y \rangle - F^*(y)\} \tag{95}$$

We know that dot product is commutative so we can re-write

$$\langle Kx, y \rangle = \langle x, K^T y \rangle \tag{96}$$

and in case the dot product is defined on hermitian space we can write it as

$$\langle Kx, y \rangle = \langle x, K^* y \rangle \tag{97}$$

where K^* is the adjoint conjugate of K which is more general. Going back to Eqn. 94, the equation now becomes

$$\min_{x \in X} \max_{y \in Y} \{\langle x, K^* y \rangle - F^*(y) + G(x)\} \tag{98}$$

Now by definition

$$\min_{x \in X} \{\langle x, K^* y \rangle + G(x)\} = -G^*(-K^* y) \tag{99}$$

because

$$\max_{x \in X}\{\langle x, K^*y\rangle - G(x)\} = G^*(K^*y) \tag{100}$$

$$\min_{x \in X}\{-\langle x, K^*y\rangle + G(x)\} = -G^*(K^*y) \tag{101}$$

$$\min_{x \in X}\{-\langle x, -K^*y\rangle + G(x)\} = -G^*(-K^*y) \tag{102}$$

$$\min_{x \in X}\{\langle x, K^*y\rangle + G(x)\} = -G^*(-K^*y) \tag{103}$$

Under the weak assumptions in convex analysis , min and max can be switched in Eqn. 98, the dual problem then becomes

$$\max_{y \in Y}\{-G^*(-K^*y) - F^*(y)\} \tag{104}$$

Primal Dual Gap is then defined as

$$\min_{x \in X}\{F(Kx) + G(x)\} - \max_{y \in Y}\{-G^*(-K^*y) - F^*(y)\} \tag{105}$$

For the primal-dual algorithm to be applicable, one should be able to compute the proximal mapping of F and G, defined as:

$$\text{Prox}_{\gamma F}(x) = \arg\min_{y} \frac{1}{2}||x - y||^2 + \gamma F(y) \tag{106}$$

Therefore, one can formulate the minimisation steps as

$$y^{n+1} = \text{Prox}_{\sigma F^*}(y^n + \sigma K\overline{x}^n) \tag{107}$$

$$x^{n+1} = \text{Prox}_{\tau G}(x^n - \tau K^* y^{n+1}) \tag{108}$$

$$\overline{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n) \tag{109}$$

Note that being able to compute the proximal mapping of F is equivalent to being able to compute the proximal mapping of F^*, due to Moreau's identity:

$$x = \text{Prox}_{\tau F^*}(x) + \tau \text{Prox}_{F/\tau}(x/\tau) \tag{110}$$

It can be shown that if $0 \leq \theta \leq 1$ and $\sigma\tau||K||^2 < 1$, x^n converges to the minimser of the original energy function.

7.1 Premilinaries

Given **a** and **x** are column vectors where the dot product $\langle \mathbf{a}, \mathbf{x} \rangle$ can be written as

$$\langle \mathbf{a}, \mathbf{x} \rangle = \mathbf{a}^T \mathbf{x} \tag{111}$$

Then we can write the associated derivates with respect to **x** as

$$\frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a} \tag{112}$$

We will be representing a 2-D image matrix as a vector in which the elements are arranged in lexicographical order.

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} = \begin{pmatrix} a_{1,1} \\ a_{1,2} \\ \vdots \\ a_{1,n} \\ a_{m,1} \\ \vdots \\ a_{m,n} \end{pmatrix} \tag{113}$$

The divergence of matrix where elements are stacked in a vectorial fashion can be derived using the following

$$\mathrm{div} A = \begin{pmatrix} \frac{\partial}{\partial x} & 0 & \cdots 0 & 0 \cdots & \frac{\partial}{\partial y} & 0 & 0 \cdots 0 & 0 \cdots & 0 \\ 0 & \frac{\partial}{\partial x} & \cdots 0 & 0 \cdots & 0 & \frac{\partial}{\partial y} & 0 \cdots 0 & \cdots & 0 \\ \vdots & \vdots & \ddots \vdots & \vdots \ddots & \vdots & \vdots & \vdots \ddots \vdots & \ddots & \vdots \\ 0 & 0 & \cdots 0 & 0 \cdots & \frac{\partial}{\partial x} & 0 & 0 \cdots 0 & \cdots & \frac{\partial}{\partial y} \end{pmatrix} \begin{pmatrix} a^x_{1,1} \\ a^x_{1,2} \\ \vdots \\ a^x_{1,n} \\ a^x_{m,1} \\ \vdots \\ a^x_{m,n} \\ a^y_{1,1} \\ a^y_{1,2} \\ \vdots \\ a^y_{1,n} \\ a^y_{m,1} \\ \vdots \\ a^y_{m,n} \end{pmatrix} \tag{114}$$

7.2 Representation of norms

We will be using the following notational brevity to represent the L_1 norm:

$$E_{tv}(u) = ||\nabla u||_1 \tag{115}$$

$$||\nabla u||_1 = \sum_{i=1}^{W} \sum_{j=1}^{H} |\nabla u_{i,j}| \tag{116}$$

$$|\nabla u_{i,j}| = \sqrt{(\partial_x u_{i,j})^2 + (\partial_y u_{i,j})^2} \tag{117}$$

where the partial derivatives are defined on discrete 2D grid as follows

$$\partial_x u_{i,j} = u(i,j) - u(i-1,j) \tag{118}$$

$$\partial_y u_{i,j} = u(i,j) - u(i,j-1) \tag{119}$$

For **Maximisation** we will be using gradient as

$$\frac{p^{n+1} - p^n}{\sigma_p} = \nabla_p E(p) \tag{120}$$

For **Minimisation** we will be using gradient as

$$\frac{p^n - p^{n+1}}{\sigma_p} = \nabla_p E(p) \tag{121}$$

NOTE the switch in the iteration numbers in Maximisation and Minimisation. For brevity, $u_{i,j}$ is denoted by u.

7.3 ROF Model

A standard ROF model can be written as

$$\min_{u \in X} ||\nabla u||_1 + \frac{\lambda}{2}||u - g||_2^2 \tag{122}$$

We know that the convex conjugate of $||.||$ norm is an indicator function

$$\delta(p) = \begin{cases} 0 & \text{if } ||p|| \leq 1 \\ \infty & \text{otherwise} \end{cases}$$

$$\therefore \quad ||\nabla u||_1 = \max_{p \in P} \Big(\langle p, \nabla u \rangle - \delta_P(p) \Big) \tag{123}$$

Therefore, we can write the ROF function as

$$\min_{u \in X} \max_{p \in P} \langle p, \nabla u \rangle + \frac{\lambda}{2}||u - g||_2^2 - \delta_P(p) \tag{124}$$

Let us call this new function $E(u, p)$

1. Compute the derivative with respect to p i.e. $\partial_p E(u, p)$ which is

$$\partial_p E(u, p) = \partial_p \Big(\langle p, \nabla u \rangle + \frac{\lambda}{2}||u - g||_2^2 - \delta_P(p) \Big) \tag{125}$$

$$\partial_p \Big(\langle p, \nabla u \rangle \Big) = \nabla u \quad \text{[proof given]} \tag{126}$$

$$\partial_p \Big(\frac{\lambda}{2}||u - g||_2^2 \Big) = 0 \tag{127}$$

$$\partial_p \delta_P(p) = 0 \qquad [\because \text{indicator function i.e. constant function}] \tag{128}$$

$$\Rightarrow \quad \partial_p E(u, p) = \nabla u \tag{129}$$

2. Compute the derivate with respect to u i.e. $\partial_u E(u,p)$ which is

$$\partial_u E(u,p) = \partial_u \Big(\langle p, \nabla u \rangle + \frac{\lambda}{2} ||u-g||_2^2 - \delta_P(p) \Big) \tag{130}$$

$$\partial_u \Big(\langle p, \nabla u \rangle \Big) = \partial_u \Big(-\langle u, \mathrm{div} p \rangle \Big) = -\mathrm{div} p \tag{131}$$

$$\partial_u \Big(\frac{\lambda}{2} ||u-g||_2^2 \Big) = \lambda(u-g) \tag{132}$$

$$\partial_u \delta_P(p) = 0 \tag{133}$$

$$\Rightarrow \quad \partial_u E(u,p) = -\mathrm{div} p + \lambda(u-g) \tag{134}$$

3. Use simple gradient descent

$$\frac{p^{n+1} - p^n}{\sigma} = \partial_p E(u,p) = \nabla u^n \tag{135}$$

$$p^{n+1} = p^n + \sigma \nabla u^n \tag{136}$$

$$p^{n+1} = \frac{p^n + \sigma \nabla u^n}{\max(1, |p^n + \sigma \nabla u^n|)} \tag{137}$$

$$\frac{u^n - u^{n+1}}{\tau} = \partial_u E(u,p) = -\mathrm{div} p + \lambda(u^{n+1} - g) \tag{138}$$

$$u^{n+1} = \frac{u^n + \tau \mathrm{div} p^{n+1} + \tau \lambda g}{1 + \tau \lambda} \tag{139}$$

7.4 Huber-ROF model

The interesting thing about Huber model is that it has a continuous first derivative, so a simple gradient descent on the function can bring us to the minima while if we were to use Newton-Raphson method which requires the second order derivative, it wouldn't be possible to do so because the second derivative of Huber model is not continuous. So, the function we want to minimise is

$$\min_{u \in X} ||\nabla u||_\alpha^h + \frac{\lambda}{2} ||u-g||_2^2 \tag{140}$$

where $||.||_h$ is the Huber-Norm and is defined as

$$||x||_\alpha = \begin{cases} \frac{|x|^2}{2\alpha} & \text{if } |x| \leq \alpha \\ |x| - \frac{\alpha}{2} & \text{if } |x| > \alpha \end{cases}$$

The convex conjugate of a parablic function can be written as

$$f^*(p) = \frac{\alpha}{2} ||p||_2^2 \quad \forall ||p|| \leq \alpha \tag{141}$$

and the conjugate of $||.||$ function is the same indicator function

$$f^*(p) = \begin{cases} \frac{\alpha}{2} & \text{if } \alpha < ||p|| \leq 1 \\ \infty & \text{otherwise} \end{cases} \tag{142}$$

Therefore the minimisation can be re-written as

$$\min_{u \in X} \max_{p \in P} \Big(\langle p, \nabla u \rangle - \delta_P(p) - \frac{\alpha}{2}||p||^2 + \frac{\lambda}{2}||u-g||_2^2 \Big) \tag{143}$$

Minimisation The minimisation can be carried out following the series of steps

1. Compute the derivative with respect to p i.e. $\partial_p E(u,p)$ which is

$$\partial_p E(u,p) = \partial_p \Big(\langle p, \nabla u \rangle - \delta_P(p) - \frac{\alpha}{2}||p||^2 + \frac{\lambda}{2}||u-g||_2^2 \Big) \tag{144}$$

$$\partial_p \Big(\langle p, \nabla u \rangle \Big) = \nabla u \tag{145}$$

$$\partial_p \Big(\frac{\lambda}{2}||u-g||_2^2 \Big) = 0 \tag{146}$$

$$\partial_p \Big(\delta_P(p) \Big) = 0 \qquad [\because \text{ p is an indicator function}] \tag{147}$$

$$\partial_p \Big(\frac{\alpha}{2}||p||^2 \Big) = \alpha p \tag{148}$$

$$\Rightarrow \quad \partial_p E(u,p) = \nabla u - \alpha p \tag{149}$$

2. Compute the derivate with respect to u i.e. $\partial_u E(u,p)$ which is

$$\partial_u E(u,p) = \partial_u \Big(\langle p, \nabla u \rangle - \delta_P(p) - \frac{\alpha}{2}||p||^2 + \frac{\lambda}{2}||u-g||_2^2 \Big) \tag{150}$$

$$\partial_u \Big(\langle p, \nabla u \rangle \Big) = \partial_u \Big(- \langle u, \mathrm{div} p \rangle \Big) = -\mathrm{div} p \tag{151}$$

$$\partial_u \Big(\frac{\lambda}{2}||u-g||_2^2 \Big) = \lambda(u-g) \tag{152}$$

$$\partial_u \delta_P(p) = 0 \tag{153}$$

$$\partial_u \Big(\frac{\alpha}{2}||p||^2 \Big) = 0 \tag{154}$$

$$\Rightarrow \quad \partial_u E(u,p) = -\mathrm{div} p + \lambda(u-g) \tag{155}$$

3. Use simple gradient descent

$$\frac{p^{n+1}-p^n}{\sigma} = \partial_p E(u,p) = \nabla u^n - \alpha p^{n+1} \tag{156}$$

$$p^{n+1} = \frac{p^n + \sigma \nabla u^n}{1+\sigma\alpha} \tag{157}$$

$$p^{n+1} = \frac{\frac{p^n+\sigma\nabla u^n}{1+\sigma\alpha}}{\max(1, |\frac{p^n+\sigma\nabla u^n}{1+\sigma\alpha}|)} \tag{158}$$

$$\frac{u^n - u^{n+1}}{\tau} = \partial_u E(u,p) = -\mathrm{div} p + \lambda(u^{n+1}-g) \tag{159}$$

$$u^{n+1} = \frac{u^n + \tau \mathrm{div} p^{n+1} + \tau\lambda g}{1+\tau\lambda} \tag{160}$$

8 TVL_1 denoising

TVL_1 denoising can be rewritten as

$$\min_{u \in X} ||\nabla u||_1 + \lambda ||u - f||_1 \tag{161}$$

This can be further rewritten as a new equation where λ is subsumed inside the norm i.e.

$$\min_{u \in X} ||\nabla u||_1 + ||\lambda(u - f)||_1 \tag{162}$$

We know that the convex conjugate of $||.||$ norm is an indicator function

$$\delta(p) = \begin{cases} 0 & \text{if } ||p|| \leq 1 \\ \infty & \text{otherwise} \end{cases}$$

$$\therefore \quad ||\nabla u||_1 = \max_{p \in P} \Big(\langle p, \nabla u \rangle - \delta_P(p) \Big) \tag{163}$$

$$and \quad ||\lambda(u - f)||_1 = \max_{q \in Q} \Big(\langle q, \lambda(u - f) \rangle - \delta_Q(q) \Big) \tag{164}$$

Therefore, we can write the TVL_1 denoising function as

$$\min_{u \in X} \max_{p \in P} \max_{q \in Q} \langle p, \nabla u \rangle + \langle q, \lambda(u - f) \rangle - \delta_P(p) - \delta_Q(q) \tag{165}$$

Let us call this new function $E(u, p, q)$

1. Compute the derivative with respect to p i.e. $\partial_p E(u, p, q)$ which is

$$\partial_p E(u, p, q) = \partial_p \Big(\langle p, \nabla u \rangle + \langle q, \lambda(u - f) \rangle - \delta_P(p) - \delta_Q(q) \Big) \tag{166}$$

$$\partial_p \Big(\langle p, \nabla u \rangle \Big) = \nabla u \quad \text{[proof given]} \tag{167}$$

$$\partial_p \Big(\langle q, \lambda(u - f) \rangle \Big) = 0 \tag{168}$$

$$\partial_p \delta_P(p) = 0 \qquad [\because \text{indicator function i.e. constant function}] \tag{169}$$

$$\partial_p \delta_Q(q) = 0 \tag{170}$$

$$\Rightarrow \quad \partial_p E(u, p, q) = \nabla u \tag{171}$$

2. Compute the derivate with respect to q i.e. $\partial_u E(u, p, q)$ which is

$$\partial_q E(u, p, q) = \partial_q \Big(\langle p, \nabla u \rangle + \langle q, \lambda(u - f) \rangle - \delta_P(p) - \delta_Q(q) \Big) \tag{172}$$

$$\partial_q \Big(\langle p, \nabla u \rangle \Big) = 0 \tag{173}$$

$$\partial_q \Big(\langle q, \lambda(u - f) \rangle \Big) = \lambda(u - f) \tag{174}$$

$$\partial_q \delta_P(p) = 0 \tag{175}$$

$$\partial_q \delta_Q(q) = 0 \tag{176}$$

$$\Rightarrow \quad \partial_q E(u, p, q) = \lambda(u - f) \tag{177}$$

3. Compute the derivate with respect to u i.e. $\partial_u E(u, p, q)$ which is

$$\partial_u E(u, p, q) = \partial_u \Big(\langle p, \nabla u \rangle + \langle q, \lambda(u - f) \rangle - \delta_P(p) - \delta_Q(q) \Big) \tag{178}$$

$$\partial_u \Big(\langle p, \nabla u \rangle \Big) = \partial_u \Big(- \langle u, \mathrm{div} p \rangle \Big) = -\mathrm{div} p \tag{179}$$

$$\partial_u \Big(\langle q, \lambda(u - f) \rangle \Big) = \lambda q \tag{180}$$

$$\partial_u \delta_P(p) = 0 \tag{181}$$

$$\partial_u \delta_Q(q) = 0 \tag{182}$$

$$\Rightarrow \quad \partial_u E(u, p) = -\mathrm{div} p + \lambda q \tag{183}$$

4. Use simple gradient descent

$$\frac{p^{n+1} - p^n}{\sigma} = \partial_p E(u, p, q) = \nabla u^n \tag{184}$$

$$p^{n+1} = p^n + \sigma \nabla u^n \tag{185}$$

$$p^{n+1} = \frac{p^n + \sigma \nabla u^n}{\max(1, |p^n + \sigma \nabla u^n|)} \tag{186}$$

$$\frac{q^{n+1} - q^n}{\sigma} = \partial_q E(u, p, q) = \lambda(u^n - f) \tag{187}$$

$$q^{n+1} = q^n + \sigma \lambda (u^n - f) \tag{188}$$

$$q^{n+1} = \frac{q^n + \sigma \lambda (u^n - f)}{\max(1, |q^n + \sigma \lambda (u^n - f)|)} \tag{189}$$

$$\frac{u^n - u^{n+1}}{\tau} = \partial_u E(u, p, q) = -\mathrm{div} p^{n+1} + \lambda q^{n+1} \tag{190}$$

$$u^{n+1} = u^n + \tau \mathrm{div} p^{n+1} - \tau \lambda q^{n+1} \tag{191}$$

8.1 Image Deconvolution

$$\min_{u \in X} ||\nabla u||_1 + \frac{\lambda}{2} ||Au - g||_2^2 \tag{192}$$

The problem can be written in terms of saddle-point problem as

$$\min_{u \in X} \max_{p \in P} \langle p, \nabla u \rangle + \frac{\lambda}{2} ||Au - g||_2^2 - \delta_P(p) \tag{193}$$

Minimisation The minimisation can be carried out following the series of steps

1. Compute the derivative with respect to p i.e. $\partial_p E(u,p)$ which is

$$\partial_p E(u,p) = \partial_p \Big(\langle p, \nabla u \rangle - \delta_P(p) + \frac{\lambda}{2}||Au-g||_2^2 \Big) \tag{194}$$
$$\partial_p \Big(\langle p, \nabla u \rangle \Big) = \nabla u \tag{195}$$
$$\partial_p \Big(\frac{\lambda}{2}||Au-g||_2^2 \Big) = 0 \tag{196}$$
$$\partial_p \Big(\delta_P(p) \Big) = 0 \qquad [\because \text{ p is an indicator function}] \tag{197}$$
$$\Rightarrow \quad \partial_p E(u,p) = \nabla u \tag{198}$$

2. Compute the derivate with respect to u i.e. $\partial_u E(u,p)$ which is

$$\partial_u E(u,p) = \partial_u \Big(\langle p, \nabla u \rangle - \delta_P(p) + \frac{\lambda}{2}||Au-g||_2^2 \Big) \tag{199}$$
$$\partial_u \Big(\langle p, \nabla u \rangle \Big) = \partial_u \Big(- \langle u, \mathrm{div} p \rangle \Big) = -\mathrm{div} p \tag{200}$$
$$\partial_u \Big(\frac{\lambda}{2}||Au-g||_2^2 \Big) = \lambda \Big(A^T A u - A^T g \Big) \tag{201}$$
$$\partial_u \delta_P(p) = 0 \tag{202}$$
$$\Rightarrow \quad \partial_u E(u,p) = -\mathrm{div} p + \lambda \Big(A^T A u - A^T g \Big) \tag{203}$$

$$\partial_u ||Au-g||_2^2 = \partial_u (Au-g)^T(Au-g) \tag{204}$$
$$(Au-g)^T(Au-g) = ((Au)^T - g^T)(Au-g) \tag{205}$$
$$((Au)^T - g^T)(Au-g) = (u^T A^T - g^T)(Au-g) \tag{206}$$
$$(u^T A^T - g^T)(Au-g) = u^T A^T A u - u^T A^T g + g^T A u + g^T g \tag{207}$$
$$\partial_u u^T A^T A u = \partial_u u^T B u \qquad [\text{Lets say B is } A^T A] \tag{208}$$
$$\partial_u u^T B u = (B + B^T) u \tag{209}$$
$$\partial_u u^T A^T A u = (A^T A + (A^T A)^T) u \tag{210}$$
$$\partial_u u^T A^T A u = (A^T A + A^T A) u \tag{211}$$
$$\partial_u u^T A^T A u = 2 A^T A u \tag{212}$$

3. Use simple gradient descent

$$\frac{p^{n+1} - p^n}{\sigma} = \partial_p E(u,p) = \nabla u^n \tag{213}$$

$$p^{n+1} = p^n + \sigma \nabla u^n \tag{214}$$

$$p^{n+1} = \frac{p^n + \sigma \nabla u^n}{\max(1, |p^n + \sigma \nabla u^n|)} \tag{215}$$

$$\frac{u^n - u^{n+1}}{\tau} = \partial_u E(u,p) = -\mathrm{div} p^{n+1} + \lambda \left(\frac{(A^T A)}{1} u^{n+1} - A^T g \right) \tag{216}$$

$$u^{n+1} \left(I + \tau \lambda A^T A \right) = u^n + \tau \mathrm{div} p^{n+1} + \tau \lambda A^T g \tag{217}$$

$$u^{n+1} = \left(I + \tau \lambda A^T A \right)^{-1} (u^n + \tau \mathrm{div} p^{n+1} + \tau \lambda A^T g) \tag{218}$$

$$\tag{219}$$

This requires matrix inversion. In some cases the matrix may be singular because it is generally sparse and therefore inversion is not a feasible solution. Therefore, one resorts to using Fourier Analysis.

Another alternative is to dualise again with respect to u, which then yields

$$\min_{u \in X} \max_{p \in P, q \in Q} \langle p, \nabla u \rangle + \langle Au - g, q \rangle - \delta_P(p) - \frac{1}{2\lambda} ||q||^2 \tag{220}$$

1. Compute the derivative with respect to p i.e. $\partial_p E(u,p,q)$ which is

$$\partial_p E(u,p) = \nabla u \tag{221}$$

2. Compute the derivate with respect to q i.e. $\partial_q E(u,p,q)$ which is

$$\partial_q E(u,p,q) = Au - g - \frac{1}{\lambda} q \tag{222}$$

3. Compute the derivate with respect to u i.e. $\partial_u E(u,p,q)$ which is

$$\partial_u E(u,p,q) = -\mathrm{div} p + A^T q \tag{223}$$

4. Use simple gradient descent

$$\frac{p^{n+1} - p^n}{\sigma_p} = \partial_p E(u, p, q) = \nabla u^n \tag{224}$$

$$p^{n+1} = p^n + \sigma_p \nabla u^n \tag{225}$$

$$p^{n+1} = \frac{p^n + \sigma_p \nabla u^n}{\max(1, |p^n + \sigma_p \nabla u^n|)} \tag{226}$$

$$\frac{q^{n+1} - q^n}{\sigma_q} = \partial_q E(u, p, q) = Au^n - g - \frac{1}{\lambda} q^{n+1} \tag{227}$$

$$q^{n+1} = \frac{q^n + \sigma_q A u^n - \sigma_q g}{1 + \frac{\sigma_q}{\lambda}} \tag{228}$$

$$\frac{u^n - u^{n+1}}{\tau} = \partial_u E(u, p) = -\mathrm{div} p^{n+1} + A^T q^{n+1} \tag{229}$$

$$u^{n+1} = u^n + \tau \mathrm{div} p^{n+1} - \tau A^T q^{n+1} \tag{230}$$

This saves matrix inversion. One of the benefits of using the Legendre-Fenchel transformation.

Interesting tip Imagine we have a function of the form

$$E = (h * u - f)^2 \tag{231}$$

where $*$ operator denotes the convolution. If one wants to take the derivate with respect to u, one can make use of the fact that $h * u$ can be expressed as a linear function of sparse matrix D, i.e. Du. Rewriting the equation we can derive

$$E = (Du - f)^2 = (Du - f)^T (Du - f) \tag{232}$$

Now it is very trivial to see the derivative of this function with respect to u. Referring to the Eqn. 76 in the [2], we can then write the derivative of E with respect to u as follows

$$\frac{\partial E}{\partial u} = 2D^T (Du - f) \tag{233}$$

$$Du = h * u \tag{234}$$

$$D^T (Du - f) = \tilde{h} * (h * u - f) \tag{235}$$

where $\tilde{h}$ is the mirrored kernel, i.e. $\tilde{h} \equiv h(-x)$

8.2 Optic Flow

Optic flow was popularised by Horn and Schunk's seminal paper [4] which has over the next two decades sparked a great interest in minimising the energy function associated with computing optic flow and it's various different formulations [5][6][9]. Writing the standard L_1 norm based optic flow equation

$$\min_{u \in X, v \in Y} \left\{ ||\nabla u||_1 + ||\nabla v||_1 + \lambda |I_1(x+f) - I_2(x)| \right\} \tag{236}$$

where f is a flow vector (u, v) at any pixel (x, y) in the image. For brevity (x, y) is replaced by x. Substituting p for dual variable corresponding to ∇u, q for ∇v and r for $I_1(x+f) - I_2(x)$, we can rewrite the original energy formulation in it's primal-dual form as

$$\begin{aligned} \max_{p \in P, q \in Q, r \in R} \min_{u \in X, v \in Y} \Big\{ \langle p, \nabla u \rangle + \langle q, \nabla v \rangle \\ + \langle r, \lambda (I_1(x+f) - I_2(x)) \rangle - \delta_p(P) - \delta_q(Q) - \delta_r(R) \Big\} \end{aligned} \tag{237}$$

We have used the same trick for writing the dual formulation corresponding to the $I_1(x+f) - I_2(x)$ by subsuming the λ inside, we used while writing the dual formulation of the data term in TV-L_1 denoising equation 164. Various derivates required for gradient descent can be computed as shown below

1. Compute the derivate with respect to p i.e. $\partial_p E(u, v, p, q, r)$ which is

$$\partial_p E(u, v, p, q, r) = \nabla u \tag{238}$$

2. Compute the derivate with respect to q i.e. $\partial_q E(u, v, p, q, r)$ which is

$$\partial_q E(u, v, p, q, r) = \nabla v \tag{239}$$

3. Compute the derivate with respect to r i.e. $\partial_r E(u, v, p, q, r)$ which is

$$\partial_r E(u, v, p, q, r) = \lambda (I_1(x+f) - I_2(x)) \tag{240}$$

4. Compute the derivate with respect to u i.e. $\partial_r E(u, v, p, q, r)$ which is

$$\partial_u E(u, v, p, q, r) = -\text{div} p + \partial_u (\langle r, \lambda (I_1(x+f) - I_2(x)) \rangle) \tag{241}$$

Linearising around f_0, we can rewrite the above expression involving r as

$$\langle r, \lambda (I_1(x+f) - I_2(x)) \rangle = \langle r, \lambda (I_1(x+f_0) - I_2(x)) + (f - f_0)^t [\mathbf{I_x} \mathbf{I_y}]^T) \rangle. \tag{242}$$

We can expand the terms involving f and f_0 as

$$I_1(x+f_0) - I_2(x) + (f - f_0)^t [\mathbf{I_x} \mathbf{I_y}]^T = I_1(x+f_0) - I_2(x) + (u - u_0)^t \mathbf{I_x} + (v - v_0)^t \mathbf{I_y} \tag{243}$$

It is easy to then expand the dot-product expression involving r as

$$\begin{aligned} &\langle r, \lambda (I_1(x+f_0) - I_2(x) + (f - f_0)^t [\mathbf{I_x} \mathbf{I_y}]^T) \rangle \\ &= \lambda \langle r, I_1(x+f_0) - I_2(x) \rangle + \lambda \langle r, (u - u_0)^t \mathbf{I_x} \rangle + \lambda \langle r, (v - v_0)^t \mathbf{I_y} \rangle \end{aligned} \tag{244}$$

$$\langle r, (u-u_0)^t \mathbf{I_x} \rangle = \langle r, \mathbf{I_x}(u-u_0) \rangle = r^t \mathbf{I_x}(u-u_0) = \langle \mathbf{I_x}^T r, (u-u_0) \rangle \quad (245)$$

$\mathbf{I_x}$ is a diagonal matrix composed of entries corresponding to gradient along x-axis for each pixel and similarly $\mathbf{I_y}$ is composed of gradient along y-axis. The derivative of E with respect to u can be then written as

$$\partial_u E(u, v, p, q, r) = -\mathrm{div} p + \lambda \mathbf{I_x}^T r \quad (246)$$

5. Compute the derivate with respect to v i.e. $\partial_v E(u, v, p, q, r)$ which is

$$\partial_v E(u, v, p, q, r) = -\mathrm{div} q + \lambda \mathbf{I_y}^T r \quad (247)$$

Gradient descent equations then follow straightforward

1. Maximise with respect to p

$$\frac{p^{n+1} - p^n}{\sigma_p} = \nabla u^n \quad (248)$$

$$p^{n+1} = \frac{p^n + \sigma_p \nabla u^n}{\max(1, |p^n + \sigma_p \nabla u^n|)} \quad (249)$$

2. Maximise with respect to q

$$\frac{q^{n+1} - q^n}{\sigma_q} = \nabla v^n \quad (250)$$

$$q^{n+1} = \frac{q^n + \sigma_q \nabla v^n}{\max(1, |q^n + \sigma_q \nabla v^n|)} \quad (251)$$

3. Maximise with respect to r

$$\frac{r^{n+1} - r^n}{\sigma_r} = \lambda(I_1(x + f^n) - I_2(x)) \quad (252)$$

$$r^{n+1} = \frac{r^n + \sigma_r \lambda(I_1(x + f^n) - I_2(x))}{\max(1, |r^n + \sigma_r \lambda(I_1(x + f^n) - I_2(x))|)} \quad (253)$$

4. Minimise with respect to u

$$\frac{u^n - u^{n+1}}{\sigma_u} = -\mathrm{div} p^{n+1} + \lambda \mathbf{I_x}^T r^{n+1} \quad (254)$$

5. Minimise with respect to v

$$\frac{v^n - v^{n+1}}{\sigma_v} = -\mathrm{div} q^{n+1} + \lambda \mathbf{I_y}^T r^{n+1} \quad (255)$$

8.3 Super-Resolution

The formulation was first used in [8] but we will describe here the minimisation procedue below.

$$\min_{u \in X} \left\{ \lambda ||\nabla \hat{u}||_{\epsilon_u}^{h} + \sum_{i=1}^{N} ||\mathbf{DBW}_i \hat{u} - \check{f}_i||_{\epsilon_d}^{\xi h} \right\} \tag{256}$$

With $\lambda > 0$, let us now rewrite the conjugate for $\lambda||.||$ we see

$$f^*(p) = \sup_{u \in \mathbb{R}} \Big(\langle p, \nabla u \rangle - \lambda ||\nabla u|| \Big) \tag{257}$$

$$f^*(p) = \lambda \sup_{u \in \mathbb{R}} \Big(\langle \frac{p}{\lambda}, \nabla u \rangle - ||\nabla u|| \Big) \tag{258}$$

Let us now denote $\frac{p}{\lambda} = k$, then we can write

$$f^*(p) = \lambda \sup_{u \in \mathbb{R}} \Big(\langle k, \nabla u \rangle - ||\nabla u|| \Big) \tag{259}$$

$$f^*(p) = \lambda f^*(k) \tag{260}$$

But we know that $\sup_{u \in \mathbb{R}} \Big(\langle k, \nabla u \rangle - ||\nabla u|| \Big)$ is an indicator function defined by

$$f^*(k) = \begin{cases} 0 & \text{if } ||k|| \leq 1 \\ \infty & \text{otherwise} \end{cases} \tag{261}$$

Therefore, we can write the $f^*(p)$ as

$$f^*(p) = \begin{cases} 0 & \text{if } ||k|| \leq 1 \\ \infty & \text{otherwise} \end{cases} \tag{262}$$

Now replace k by $\frac{p}{\lambda}$ we can then come to an expression

$$f^*(p) = \begin{cases} 0 & \text{if } ||\frac{p}{\lambda}|| \leq 1 \\ \infty & \text{otherwise} \end{cases} \tag{263}$$

The saddle-point formulation then becomes

$$\begin{aligned} \min_{u \in X} \max_{\hat{p}, \check{q}} \Bigg\{ & \langle \hat{p}, \nabla \hat{u} \rangle - \frac{\epsilon_u}{2\lambda h^2} ||\hat{p}||^2 - \delta_{\{|\hat{p}| \leq \lambda h^2\}} \\ & + \sum_{i=1}^{N} \Bigg(\langle \hat{q}_i, \mathbf{DBW}_i \hat{u} - \check{f}_i \rangle_X^{\xi h} - \frac{\epsilon_d}{2(\xi h)^2} ||\check{q}||^2 - \delta_{\{|\check{q}_i| \leq (\xi h)^2\}} \Bigg) \Bigg\} \end{aligned} \tag{264}$$

Minimisation Minimisation equations can be written as follows

1. Compute the derivative with respect to p i.e. $\partial_p E(u,p)$ which is

$$\partial_p E(\hat{u},\hat{p},\check{q}) = \partial_p \Bigg\{ \langle \hat{p}, \nabla \hat{u} \rangle - \frac{\epsilon_u}{2\lambda h^2} ||\hat{p}||^2 - \delta_{\{|\hat{p}| \le \lambda h^2\}} + \sum_{i=1}^{N} \Bigg(\langle \hat{q}_i, \mathbf{DBW}_i \hat{u} - \check{f}_i \rangle_X^{\xi h} - \frac{\epsilon_d}{2(\xi h)^2} ||\check{q}||^2 - \delta_{\{|\check{q}_i| \le (\xi h)^2\}} \Bigg) \Bigg\} \tag{265}$$

$$\partial_p \Big(\langle p, \nabla u \rangle \Big) = \nabla u \tag{266}$$

$$\Rightarrow \quad \partial_p E(u,p) = \nabla u - \frac{\epsilon_u}{\lambda h^2} \hat{p} \tag{267}$$

2. Compute the derivate with respect to q_i i.e. $\partial_u E(u,p,q_i)$ which is

$$\partial_{\check{q}_i} E(\hat{u},\hat{p},\check{q}_i) = (\xi h)^2 (\mathbf{DBW}_i \hat{u} - \check{f}_i) - \partial_{\check{q}_i} \Bigg(\frac{\epsilon_d}{2(\xi h)^2} ||\check{q}_i||^2 \Bigg) \tag{268}$$

$$\Rightarrow \quad \partial_{\check{q}_i} E(\hat{u},\hat{p},\check{q}_i) = (\xi h)^2 (\mathbf{DBW}_i \hat{u} - \check{f}_i) - \frac{\epsilon_d}{(\xi h)^2} \check{q}_i \tag{269}$$

3. Compute the derivate with respect to u i.e. $\partial_u E(u,p,q_i)$ which is

$$\partial_u E(\hat{u},\hat{p},\check{q}_i) = -\mathrm{div} p + (\xi h)^2 \sum_{i=1}^{N} \partial_u (\check{q}_i^T (\mathbf{DBW}_i \hat{u} - \check{f}_i)) \tag{270}$$

$$\Rightarrow \quad \partial_u E(\hat{u},\hat{p},\check{q}_i) = -\mathrm{div} p + (\xi h)^2 \sum_{i=1}^{N} (\mathbf{W}_i^T \mathbf{B}^T \mathbf{D}^T \check{q}_i) \tag{271}$$

4. Use simple gradient descent

$$\frac{p^{n+1} - p^n}{\sigma_p} = \partial_p E(u,p) = h^2 \nabla u^n - \frac{\epsilon_u}{\lambda h^2} p^{n+1} \tag{272}$$

$$p^{n+1} - p^n = \sigma_p h^2 \nabla^h u^n - \frac{\sigma_p \epsilon_u}{\lambda h^2} p^{n+1} \tag{273}$$

$$p^{n+1} = \frac{\sigma_p h^2 \nabla^h u^n + p^n}{1 + \frac{\sigma_p \epsilon_u}{\lambda h^2}} \tag{274}$$

$$p^{n+1} = \frac{p^{n+1}}{\max(1, \frac{|p^{n+1}|}{\lambda})} \tag{275}$$

$$\frac{\check{q}_i^{n+1} - \check{q}_i^n}{\sigma_q} = (\xi h)^2(\mathbf{DBW}_i\hat{u}^n - \check{f}_i) - \frac{\epsilon_d}{(\xi h)^2}\check{q}_i^{n+1} \tag{276}$$

$$\check{q}_i^{n+1} - \check{q}_i^n = \sigma_q(\xi h)^2(\mathbf{DBW}_i\hat{u}^n - \check{f}_i) - \frac{\sigma_q\epsilon_d}{(\xi h)^2}\check{q}_i^{n+1} \tag{277}$$

$$\check{q}_i^{n+1} = \frac{\check{q}_i^n + \sigma_q(\xi h)^2(\mathbf{DBW}_i\hat{u}^n - \check{f}_i)}{1 + \frac{\sigma_q\epsilon_d}{(\xi h)^2}} \tag{278}$$

$$\check{q}_i^{n+1} = \frac{\check{q}_i^{n+1}}{\max(1, |\check{q}_i^{n+1}|)} \tag{279}$$

$$\frac{\hat{u}^n - \hat{u}^{n+1}}{\tau} = \partial_u E(\hat{u}, \hat{p}, \check{q}_i) = -\mathrm{div}p^{n+1} + (\xi h)^2\sum_{i=1}^{N}(\mathbf{W}_i^T\mathbf{B}^T\mathbf{D}^T\check{q}_i^{n+1}) \tag{280}$$

$$\hat{u}^{n+1} = \hat{u}^n - \tau\Bigg(-\mathrm{div}p^{n+1} + (\xi h)^2\sum_{i=1}^{N}(\mathbf{W}_i^T\mathbf{B}^T\mathbf{D}^T\check{q}_i^{n+1})\Bigg) \tag{281}$$

8.4 Super Resolution with Joint Flow Estimation

Let us now try to turn our attention towards doing full joint tracking and super resolution image reconstruction. Before we derive anything let's try to formulate the problem from bayesian point of view. We are given the downsampling, blurring operators and we want to determine the optical flow between the images and reconstruct the super resolution image at the same time. The posterior probability can be written as

$$P(\hat{u}, \{\hat{w}_i\}_{i=1}^N|\{\check{f}_i\}_{i=1}^N, D, B) \tag{282}$$

Using standard bayes rule, we can write this in terms of likelihoods and priors as

$$P(\hat{u}, \{\hat{w}_i\}_{i=1}^N|\{\check{f}_i\}_{i=1}^N, D, B) \propto \prod_{i=1}^{N} P(\check{f}_i|\hat{w}_i, \hat{u}, D, B)P(\hat{w}_i, \hat{u}, D, B) \tag{283}$$

$P(\check{f}_i|\hat{w}_i, \hat{u}, D, B)$ is our standard super resolution likelihood model and under the L_1 norm can be expressed as follows

$$-\log P(\check{f}_i|\hat{w}_i, \hat{u}, D, B) = ||DB\hat{u}(x + \hat{w}_i) - \check{f}_i|| \tag{284}$$

while $P(\hat{w}_i, \hat{u}, D, B)$ marks our prior for the super resolution image and the flow. It can be easily simplied under the assumption that flow prior is independent of super resolution prior.

$$P(\hat{w}_i, \hat{u}, D, B) = \Bigg\{\prod_{i=1}^{N} P(\hat{w}_i)\Bigg\}P(\hat{u}) \tag{285}$$

The priors are standard TV-L_1 priors and can be written as

$$-\log P(\hat{w}_i) = \mu_i\{||\nabla\hat{w}_{xi}|| + ||\nabla\hat{w}_{yi}||\} \tag{286}$$

and

$$-\log P(\hat{u}) = \lambda||\nabla\hat{u}|| \tag{287}$$

The combined energy function can be written as

$$E(\hat{u}, \{\hat{w}_i\}_{i=1}^N) = \sum_{i=1}^N ||DB\hat{u}(x+\hat{w}_i) - \check{f}_i|| + \sum_{i=1}^N \mu_i\{||\nabla\hat{w}_{xi}|| + ||\nabla\hat{w}_{yi}||\} + \lambda||\nabla\hat{u}|| \tag{288}$$

We dualise with respect to each L_1 norm and obtain the following expression

$$\begin{aligned} E(\hat{u}, \{\hat{w}_i\}_{i=1}^N) = &\sum_{i=1}^N \langle q_i, DB\hat{u}(x+\hat{w}_i) - \check{f}_i\rangle \\ &+ \sum_{i=1}^N \mu_i\{\langle r_{xi}, \nabla\hat{w}_{xi}\rangle + \langle r_{yi}, \nabla\hat{w}_{yi}\rangle\} + \lambda\langle p, \nabla\hat{u}\rangle \end{aligned} \tag{289}$$

Optimising with respect to q_i

$$\frac{q_i^{n+1} - q_i^n}{\sigma_q} = DB\hat{u}^n(x + w_i^n) - \check{f}_i \tag{290}$$

Optimising with respect to p

$$\frac{p^{n+1} - p^n}{\sigma_p} = \nabla u^n \tag{291}$$

Optimising with respect to r_{xi}

$$\frac{r_{xi}^{n+1} - r_{xi}^n}{\sigma_{r_x}} = \nabla w_{xi}^n \tag{292}$$

Optimising with respect to r_{yi}

$$\frac{r_{yi}^{n+1} - r_{yi}^n}{\sigma_{r_y}} = \nabla w_{yi}^n \tag{293}$$

Optimising with respect to $\hat{w}_{xi}$ The linearisation around the current solution leads to expanding the flow equation as

$$DB\hat{u}(x+\hat{w}_i^n) - \check{f}_i = DB\hat{u}(x+\hat{w}_i^{n-1} + \hat{dw}_i^n) - \check{f}_i \tag{294}$$

$$\tag{295}$$

$$\begin{aligned} DB\hat{u}(x+\hat{w}_i^{n-1}+\hat{dw}_i^n) - \check{f}_i = DB\{&\hat{u}(x+\hat{w}_i^{n-1}) + \partial_x\hat{u}(x+\hat{w}_i^{n-1})\hat{dw}_{xi}^n \\ &+ \partial_y\hat{u}(x+\hat{w}_i^{n-1})\hat{dw}_{yi}^n\} - \check{f}_i \end{aligned} \tag{296}$$

Replacing $\hat{dw}_{xi}^n$ and $\hat{dw}_{yi}^n$ by $\hat{w}_{xi}^n - \hat{w}_{xi}^{n-1}$ and $\hat{w}_{yi}^n - \hat{w}_{yi}^{n-1}$ respectively we can rewrite the above equation as

$$\begin{aligned} DB\hat{u}(x+\hat{w}_i^{n-1}+\hat{dw}_i^n) - \check{f}_i = DB\{&\hat{u}(x+\hat{w}_i^{n-1}) + \partial_x\hat{u}(x+\hat{w}_i^{n-1})(\hat{w}_{xi}^n - \hat{w}_{xi}^{n-1}) \\ &+ \partial_y\hat{u}(x+\hat{w}_i^{n-1})(\hat{w}_{yi}^n - \hat{w}_{yi}^{n-1})\} - \check{f}_i \end{aligned} \tag{297}$$

Treating now $\hat{w}_i^{n-1}$ to be constant, we can minimise the energy function with respect to $\hat{w}_{xi}$ and $\hat{w}_{yi}^n$ respectively. The obtained update equations can be written as

$$\begin{aligned} \frac{\hat{w}_{xi}^n - \hat{w}_{xi}^{n-1}}{\sigma_w} = \partial_{\hat{w}_{xi}}\Bigg\{\langle q_i, DB\Bigg\{&\hat{u}(x+\hat{w}_i^{n-1}) + \partial_x\hat{u}(x+\hat{w}_i^{n-1})(\hat{w}_{xi}^n - \hat{w}_{xi}^{n-1}) \\ &+ \partial_y\hat{u}(x+\hat{w}_i^{n-1})(\hat{w}_{yi}^n - \hat{w}_{yi}^{n-1})\Bigg\} - \check{f}_i\rangle + \langle r_{xi}, \nabla\hat{w}_{xi}\rangle + \langle r_{yi}, \nabla\hat{w}_{yi}\rangle\Bigg\} \end{aligned} \tag{298}$$

$$\frac{\hat{w}_{xi}^n - \hat{w}_{xi}^{n-1}}{\sigma_w} = I_x^T B^T D^T q_i^n - \mathrm{div} r_{xi}^n \tag{299}$$

or

$$\frac{\hat{w}_{xi}^{n+1} - \hat{w}_{xi}^n}{\sigma_w} = I_x^T B^T D^T q_i^{n+1} - \mathrm{div} r_{xi}^{n+1} \tag{300}$$

$$I_x = \mathrm{diag}(\partial_x(\hat{u}(x+\hat{w}_i^n))) \tag{301}$$

Optimising with respect to $\hat{w}_{yi}$ Similar optimisation scheme with respect to $\hat{w}_{yi}$ yields a similar update equation

$$\frac{\hat{w}_{yi}^{n+1} - \hat{w}_{yi}^n}{\sigma_w} = I_y^T B^T D^T q_i^{n+1} - \mathrm{div} r_{yi}^{n+1} \tag{302}$$

$$I_y = \mathrm{diag}(\partial_y(\hat{u}(x+\hat{w}_i^n))) \tag{303}$$

Optimisations with respect to q_i, w_{xi}, w_{yi}, r_{xi} and r_{yi} are done on a coarse-to-fine pyramid fashion.

Optimising with respect to $\hat{u}$ Given the current solution for $\hat{w}_i^n$ we can write $\hat{u}(x + \hat{w}_i^n)$ as a linear function of $\hat{u}$ by multiplying it with warping matrix $W_i^n \hat{u}$

$$\frac{\hat{u}^{n+1} - \hat{u}^n}{\sigma_{\hat{u}}} = -\left(\sum_{i=1}^{N} (W_i^{(n+1)})^T B^T D^T q_i^{n+1} - \mathrm{div} p^{n+1} \right) \tag{304}$$

9 Setting the step sizes

The constants τ and σ are usually very easy to set if the operator K in the equation

$$\min_{x \in X} F(Kx) + G(x) \tag{305}$$

is a simple operator, in which case τ and σ can be easily found from the constraint that $\tau\, \sigma\, L^2 \leq 1$, where L is the operator norm i.e. $||K||$. For instance if we try to look at our problem of ROF model and dualise it we can see that

$$\min_{u \in X} ||\nabla u||_1 + \frac{\lambda}{2} ||u - g||_2^2 \tag{306}$$

using p as a dual to ∇u and q as a dual to $u - g$ we can reduce this to it's dual form

$$\min_{u \in X} \max_{p \in P} \max_{q \in Q} \langle p, \nabla u \rangle - \delta_P(p) + \langle q, u - f \rangle - \frac{1}{2\lambda} q^2 \tag{307}$$

Let us denote that we want to use a single dual variable y as a substitute for concatenated vector of p and q, only to simply this equation to obtain an expression in the form of Eqn. 94 so that we treat our u in this equation as x there. We can then rewrite this above expression very simply in x and y as

$$\min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle - F^*(y) + G(x) \tag{308}$$

where our K now is

$$K = \begin{pmatrix} \nabla \\ I \end{pmatrix}, x = u, y = \begin{pmatrix} p \\ q \end{pmatrix}, F^*(y) = \begin{pmatrix} \delta_P(p) \\ q^T f + \frac{1}{2\lambda} q^2 \end{pmatrix} \text{ and } G(x) = \begin{pmatrix} 0 \end{pmatrix} \tag{309}$$

It is easy to see that if K has a simple form, we can write the closed form solution of the norm of K, i.e. $||K||$. However, if K has some complicated structure e.g. in the case of deblurring or super resolution K would have different entries in each row and it's hard to come up with a closed form solution of the norm of K in which case one would like to know how to set the τ and σ so that we can carry out the succession of iterations for our variables involved in minimisation. A new formulation from Pock et al. [10] describe a way to set the τ and σ such that the optimality condition of convergence still holds. It is

$$\tau_{\mathrm{j}} = \frac{1}{\sum_{i=1}^{N} |K_{\mathrm{ij}}|^{2-\alpha}} \quad \text{and} \quad \sigma_{\mathrm{i}} = \frac{1}{\sum_{j=1}^{M} |K_{\mathrm{ij}}|^{\alpha}} \tag{310}$$

where generally $\alpha = 1$.

10 When to and when not to use Duality: What price do we pay on dualising a function?

It may at first seem a bit confusing that adding more variables using duality makes the optimisation quicker. However, expressing any convex function as a combination of simple linear functions in dual space makes the whole problem easy to handle. Working on primal and dual problems at the same time brings us closer to the solution very quickly. Switching between min and max between the optimisation means a strong duality holds.

10.1 If the function is well convex and differentiable, should we still dualise?

Let us take an example of a function which is convex and differential everywhere. We take the ROF model and replace the L_1 norm with the standard L_2 norm, i.e.

$$E(u, \nabla u) = \min_{u \in X} ||\nabla u||_2^2 + \frac{\lambda}{2}||u - g||_2^2 \tag{311}$$

If we were to use the standard Euler-Lagrange equations, we will obtain the following update equation

$$\frac{u^{n+1} - u^n}{\sigma_u} = -\frac{\partial E(u, \nabla u)}{\partial u} \tag{312}$$

where $\frac{\partial E(u,\nabla u)}{\partial u}$ is defined according to Euler-Lagrange equation as

$$\frac{\partial E(u, \nabla u)}{\partial u} = \frac{\partial E}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial E}{\partial u_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial E}{\partial u_y}\right) \tag{313}$$

where ofcourse our $||\nabla u||^2$ is defined as $(u_x^2 + u_y^2)$ where u_x is the derivative with respect to x and similarly for y. Now if we were to write the gradient descent update step with respect to u, we will obtain the following update-equation.

$$\frac{u^{n+1} - u^n}{\sigma_u} = -(\lambda(u - f) - 2\frac{\partial}{\partial x}(u_x) - 2\frac{\partial}{\partial y}(u_y)) \tag{314}$$

It therefore involves the Laplacian which is nothing but the

$$\nabla^2 u = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{315}$$

Therefore our minimisation with respect to u takes us to the final gradient step update as

$$\frac{u^{n+1} - u^n}{\sigma_u} = -(\lambda(u - f) - 2\nabla^2 u) \tag{316}$$

It is therefore clear that if we were to use the Euler-Lagrange equations, we will still have to compute the derivative of the regulariser in order to update with respect to u. However, if we were to dualise the function, we could see the data-term and regulariser-term are decoupled. Following is the Primal-Dual min-max equation for the same problem.

$$\min_{u \in X} \max_{p \in P} \max_{q \in Q} \langle p, \nabla u \rangle - \frac{1}{2} p^2 + \langle q, u - f \rangle - \frac{1}{2\lambda} q^2 \tag{317}$$

Optimising with respect to p

$$\frac{p^{n+1} - p^n}{\sigma_p} = \nabla u^n - p^{n+1} \tag{318}$$

$$p^{n+1} = \frac{p^n + \sigma_p \nabla u^n}{1 + \sigma_p} \tag{319}$$

Optimising with respect to q

$$\frac{q^{n+1} - q^n}{\sigma_q} = u^n - f - \frac{1}{\lambda} q^{n+1} \tag{320}$$

Optimising with respect to u

$$\frac{u^n - u^{n+1}}{\sigma_u} = -\mathrm{div} p^{n+1} + q^{n+1} \tag{321}$$

It is clear that while updating q we only work on $u - f$ and while updating p we only work on ∇u. The equations we obtain are system of linear equations and are pointwise separable. While in the case of Euler-Lagrange we will have to approximate the Laplacian operator with a kernel which makes the solution at on point dependent on the neighbours. Therefore, the Primal-Dual form decouples the data and the regulariser terms. It makes the problem easier to handle.

References

1. Rockafellar. T.: Convex Analysis. II
2. Matrix Cookbook: `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf` XXIII
3. `http://gpu4vision.icg.tugraz.at/`
4. B.K.P. Horn and B.G. Schunck.: Determining optical flow. Artificial Intelligence, vol 17, pp 185-203, 1981. XXIII
5. Simon Baker and Ian Matthews: Lucas-Kanade 20 Years On: A Unifying Framework, International journal of computer vision, Volume 56, Number 3, 221-255 XXIV
6. C. Zach, T. Pock and H. Bishof: A Duality Based Approach for Realtime TV-L1 Optical Flow, Proceedings of the DAGM conference on Pattern recognition, 2007. XXIV

7. Chambolle, A., Pock, T.: A First-Order Primal-Dual Algorithm for Convex Problems, Journal of Mathematical Imaging and Vision XIII
8. Unger, M., Pock, T., Werlberger, M., Bishof, H.: A covex approach for variational Super-Resolution, XXVI
9. Steinbruecker, F., Pock, T., Cremers, D.: Large Displacement Optical Flow Computation without Warping, International Conference on Computer Vision 2009 XXIV
10. Pock, T., Chambolle, A.:Diagonal preconditioning for first order primal-dual algorithms in convex optimization, International Conference on Computer Vision 2011 XXXI